

LangSmith Deployment: MCP Support For Your LangGraph Agents

6 MIN · YOUTUBE · [HTTPS://WWW.YOUTUBE.COM/WATCH?V=AR4MLBM-0RU](https://www.youtube.com/watch?v=AR4MLBM-0RU)

<https://www.youtube.com/watch?v=AR4mLbm-0RU>

SUMMARY

The video discusses the integration of the Model Context Protocol (MCP) into the Langgraph platform, enabling seamless deployment and connection of AI agents across various clients that support streamable HTTP. This integration simplifies the process of using AI agents by eliminating the need for custom code and providing a secure, standardized interface for real-time communication.

- MCP (Model Context Protocol) is an open standard for connecting large language models with external data sources and tools.*
- It addresses integration challenges by providing a single interface for all LLMs, reducing the need for custom code.*
- The Langgraph platform supports MCP, allowing developers to deploy agents accessible via MCP endpoints.*
- Deployment of agents on Langgraph is straightforward, requiring minimal configuration and no infrastructure management.*
- Each deployed agent automatically becomes an MCP server, enabling live streaming of responses without additional setup.*
- Langsmith provides built-in observability, capturing every request and response for end-to-end tracing.*
- Multiple agents can be configured and used together, facilitating complex workflows and interactions.*
- The process of deploying an agent and configuring it as an MCP tool can be completed in under five minutes.*

What if you could run your AI agent as a tool in any client that supports streamable HTTP through MCP? Well, Claude just ran our finance agent. A few lines in Claude desktop's config, zero extra code, and we configured our Langgraph agent as a live tool. First, what exactly is MCP? It stands for model context protocol and it's an open standard for directly connecting large language models with external data sources and tools. Think about the current pain of integrations.

Every LLM needing custom code to connect with every new data source. MCP solves this MM integration problem with one simple standard interface. And adoption is already strong. The MCP ecosystem encompasses thousands of tools and continues to expand as more organizations integrate it into their AI workflows. Plus, MCP provides secure two-way connection, meaning tools and models communicate safely in real time. No headaches. At Langchain, we recognize the potential of MCP to enhance AI agent capabilities. That's why we've integrated MCP support directly into the Langgraph platform. This integration allows developers to deploy agents that are immediately accessible via an MCP endpoint, facilitating easier connections with other systems and tools. Now, let's see how you can do this yourself. The first step in leveraging our agents deployed on Langgraph platform as callable MCP tools is creating a new deployment. So, we'll navigate to the deployments

tab and click new deployment. I have already authenticated my GitHub here, but if you haven't, this is where you would do it.

So, I'll navigate to finance agent. We'll give this a better name called my finance agent. We specify the git branch to look in and this langraph JSON is automatically populated for us. Optionally, we have the choice of including any resource tags. I won't for the time being. Next, we choose our deployment type. MCP is natively supported for all deployment types, so we will stay with development for the time being. Next, we'll add our environment variables. What's really nice is if you have those copied on your clipboard, you can paste them here and we will do the formatting for you. And we can see that a tracing project has been created for this deployment. Awesome. Let's go ahead and deploy. Here in the deployment screen, we can see that we are waiting for our deployment to be active. I'll check back in when it's finished. All right, our finance agent is deployed. We're on the Langraph platform deployments dashboard.

See that green tick? That means our revision is deployed and healthy. On the right, we've got our API URL for My Finance Agent. One click to spin it up. Zero infrastructure headaches. Let's click API docs. This is your one-stop shop for everything your client needs. If we scroll down and see here, there it is. Our MCP endpoint implemented exactly to the streamable HTTP spec. That means live chunk streaming of your agents responses right out of the box. No reverse proxies, no custom adapters. Langraph platform auto exposes every deployed agent as an MCP server. And look at that built-in security. You just provide the API key header and you're good to go. Plus, every request and response is captured in Langsmith Traces for end-to-end observability. In under 30 seconds, we've deployed an agent, opened its docs, and discovered a standards compliant MCP endpoint ready for any client. That's the power of Langraph platform's managed MCP support. Deploy once, connect anywhere, scale instantly.

Awesome. Now that I have my finance agent deployed, I'm going to go ahead and copy the URL that we want to use to configure our agent as an MCP tool. I'm then going to open up Claude's desktop config file. We've configured a single finance agent as our MCP server, and I can just paste in the API URL and add the MCP suffix. That's all I need. Now that we've got our agent configured as an MCP tool, let's check it out in action and visualize the tracing support that is natively built in to Langraph platform with Langsmith. As I submit my question, we can see in this window on the right the Langraph platform deployment page. We can see the recent traces populate live as we ask a question in our call client. Opening that up will show us the run tree where we can visualize our trace end to end understanding how our agent performed.

And it's that easy. Tracing is natively built in to your Langraph agents with Langsmith. Now, just as we did with the finance agent, I've gone ahead and deployed a sales agent and a marketing agent. I want to use these agents as tools in my MCP client as well. I just went ahead and updated Claude's desktop config with the API URLs of my two new agents. So, if we click on the search and tools button, we can now see our sales agent and our marketing agent registered as tools. Let's just make sure they're turned on so that we're ready to go. Great. Let's now ask a question that will engage multiple of our AI agents that we have configured as tools.

We can see that both the sales agent and the marketing agent were called, giving Claude the context it needed to

answer our question. And it's as easy as that. We've just shown how separate agents can work together as tools in any client that implements streamable HTTP for custom MCP tools. With just one click, every Langraph deployment immediately becomes an MCP endpoint. No custom service, no glue code. Langraph platform handles the heavy lifting while Langmith provides full observability, capturing every call and response. And when you're ready to scale, you can compose multiple agents like building blocks, enabling sophisticated multi-step workflows.

Ready to get started? Deploy an agent on Langraph platform and plug the MCP URL into a client like Claude. It takes just 5 minutes. No glue required. Happy building.